



Step by Step: Extending SAP Access Control with SAP Fiori Applications

Vyacheslav (Slava) Plyushchikov
Advanced View Computer Technologies

Produced by Wellesley Information Services, LLC, publisher of SAPinsider. © 2016 Wellesley Information Services. All rights reserved.



In This Session

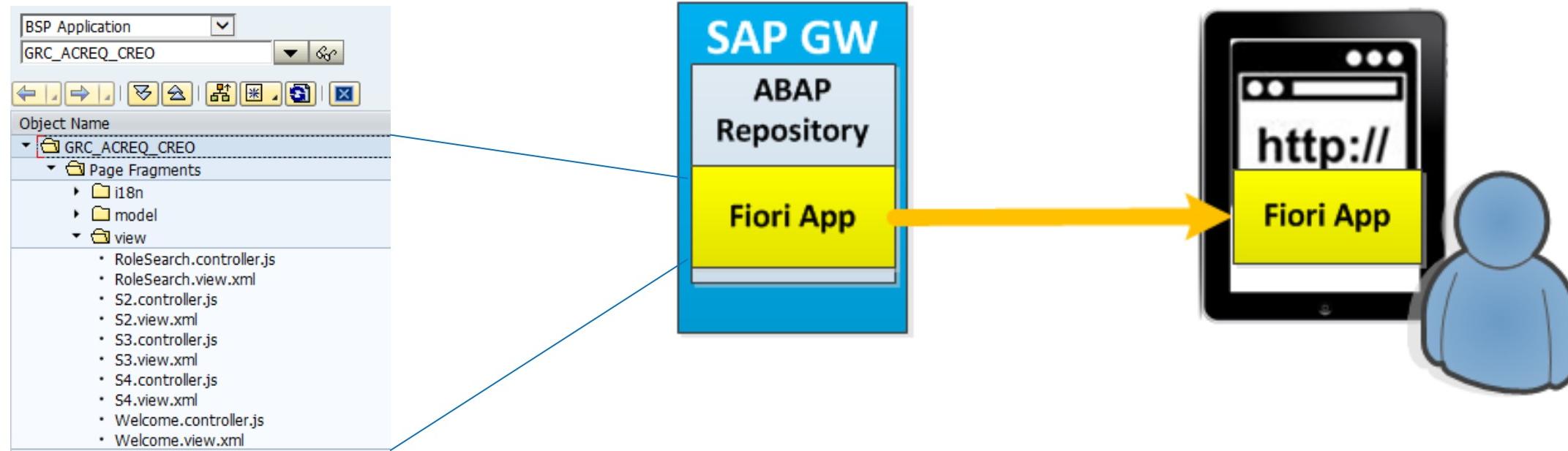
- Step-by-step process of:
 - Planning Fiori apps enhancements
 - Deploying Web IDE
 - Extending GRC Access Control Fiori apps
 - Tracing and debugging Fiori app front end
 - Testing and debugging OData calls
 - Tips and tricks of deploying extended Fiori apps

What We'll Cover

- **GRC Access Control Fiori overview**
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

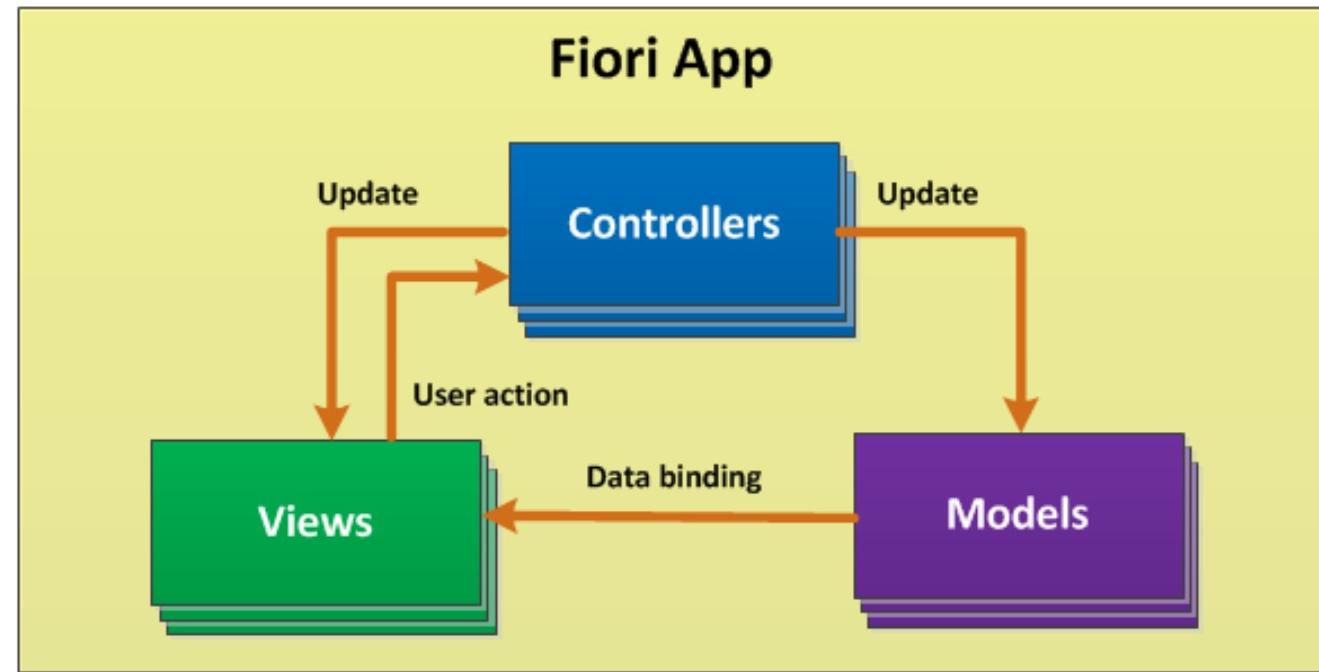
SAP Fiori: Technology

- Technology is based on HTML5/SAPUI5
- Fiori app resides in the ABAP repository as BSP application
- When executed, entire Fiori app is transferred to the HTML5-compatible web browser



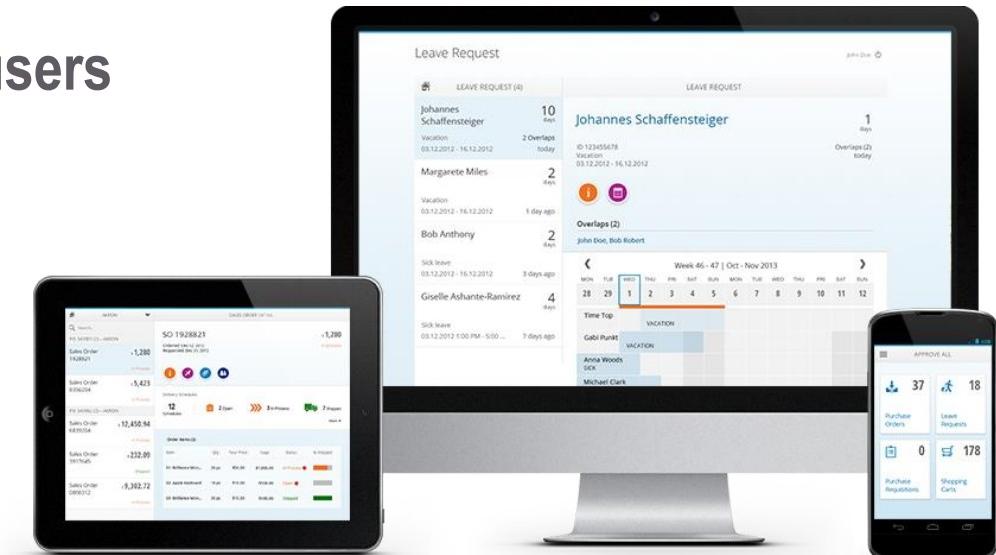
SAP Fiori: Technology (cont.)

- Model-View-Controller (MVC) concept
 - Separation between data presentation (view), flow logic (controller), and data persistence (model)
 - Better code reuse: change to one component does not affect other components



SAP Fiori: Benefits

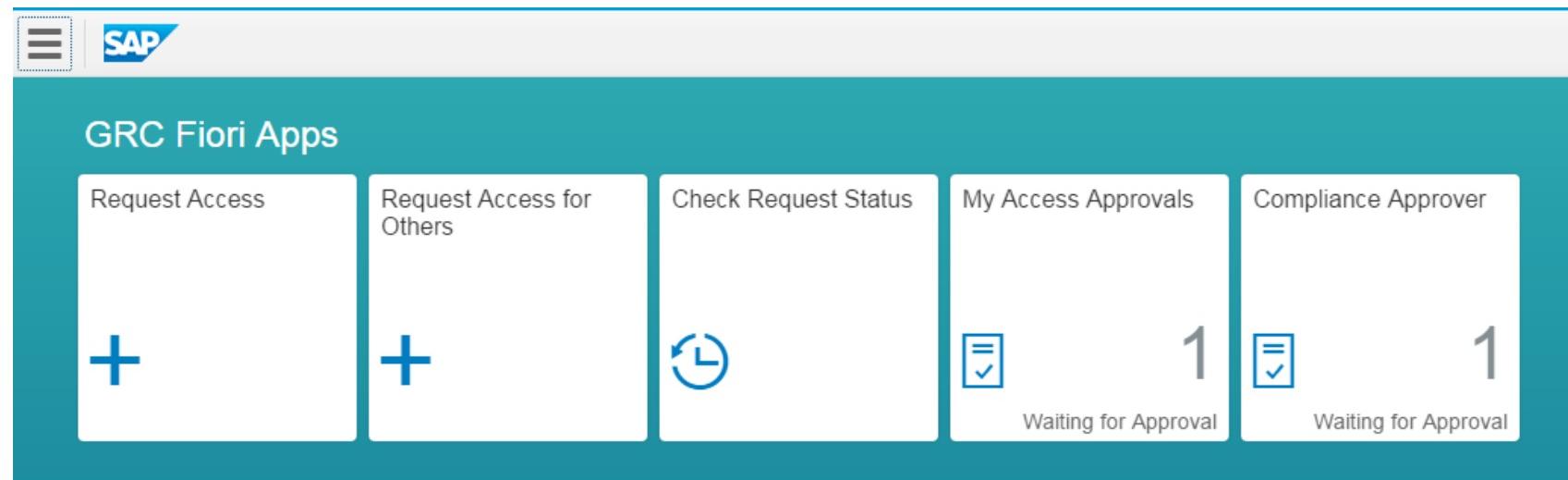
- Supported across multiple platforms, including mobile devices
 - Any device with HTML5-compatible browser can be used
- Requires zero footprint on the user device
 - No application deployment: entire app is loaded to the browser during the call
- Easy maintenance and support
 - Update on the server immediately available to all users



Source: SAP

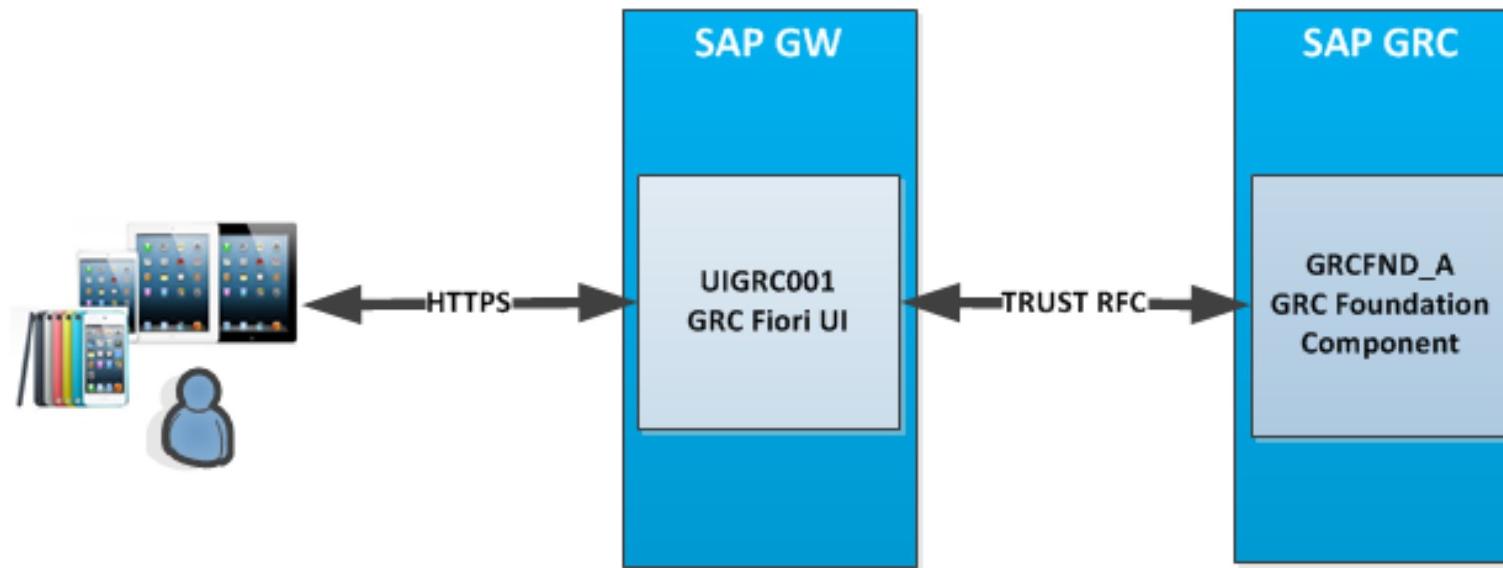
SAP GRC AC Fiori Suite

- Out-of-the-box transactional Fiori apps:
 - Request Access
 - Request Access for Others
 - Check Request Status
 - My Access Approvals
 - Compliance Approver



SAP GRC AC Fiori Suite (cont.)

- Front-end Component **UIGRC001** is deployed on SAP Gateway
- Back-end OData services delivered in **GRCFND_A** on SAP GRC Server

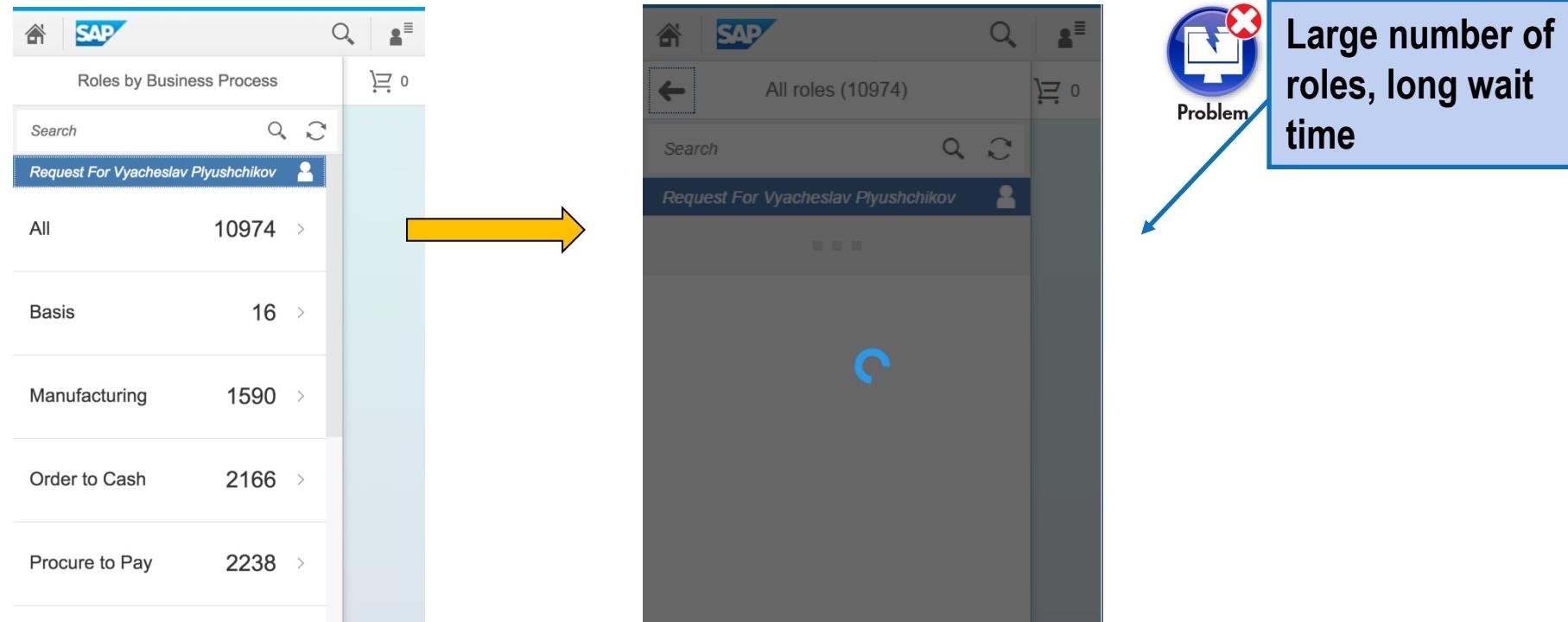


What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Business Scenario for Fiori Extension

- Issue with access request creation:
 - Company Role Catalog does not fit into out-of-the-box Fiori app functionality
 - Users cannot efficiently select roles
 - Performance issue due to large number of entries returned



Business Scenario for Fiori Extension (cont.)

- Standard app allows search only by business process or role keyword
- Company role catalog is based on employee geographical location and job function

Business Process	Role Count
PTP	2K roles
OTC	3K roles
RTR	...K roles
...	...

Each Business processes many roles. It is difficult to find the correct one.

Location	Job	Role Count
Argentina	Affiliate	5 roles
	Logistics	10 roles
	Tax	7 roles
	Treasury	5 roles

Brazil	Accounting	3 roles
	Affiliate	5 roles
	Logistics	3 roles
	Tax	7 roles

...
...

With additional filtering by Location and Job, number of roles to choose becomes reasonable

Business Scenario for Fiori Extension (cont.)

- Solution: Extend “Request Access for Others” app
 - Add required search fields
 - Allow “Expert” mode for Super Users

The diagram illustrates the SAP Fiori Request Access for Others app interface. On the left, a screenshot shows the search conditions page with a yellow box highlighting the 'Expert Mode' checkbox. A large yellow arrow points from this screen to the right. On the right, a screenshot shows the results page for 'roles (11)', with a yellow box highlighting the count 'roles (11)'. A blue line connects this box to a callout box containing the text: 'Now only 11 roles are returned. Easy for user to choose from.' The results list includes two entries:

Role	Environment	Business Process	Technical Name
BUSINESS SUPPORT FOR SCE-GLOBAL	All	Supply Chain Execution	BUSINESS SUPPORT FOR SCE-GLOBAL
BUSINESS SUPPORT-RTR-GLOBAL	All		

Now only 11 roles are returned. Easy for user to choose from.

Demo: Extended Fiori App

Extended Fiori App in Action: Request Access for Other



Demo

What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

IDE Setup

- Option 1: Cloud/SAP HCP
 - SAP Web IDE hosted on SAP HCP
 - Free trial option and subscription-based option
- Option 2: Local Web IDE installation
 - Installed locally on Windows Workstation
 - Free trial version of Web IDE 1.12

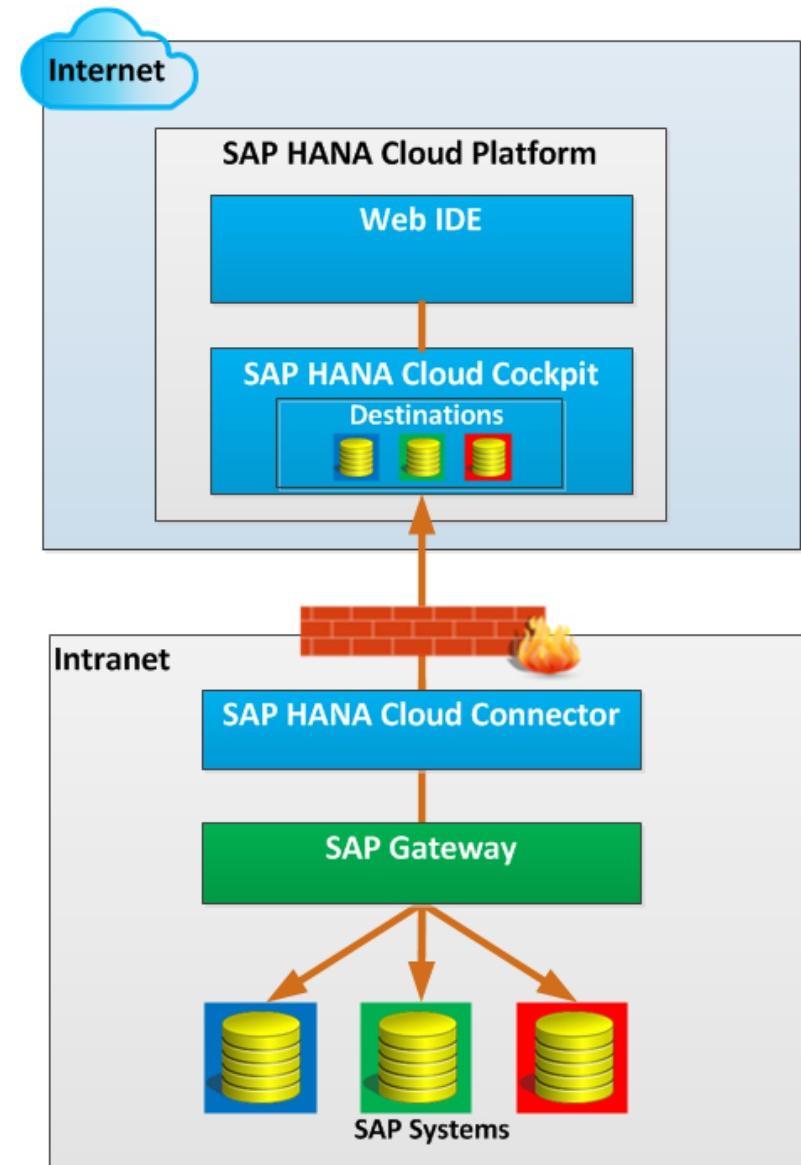


VS.



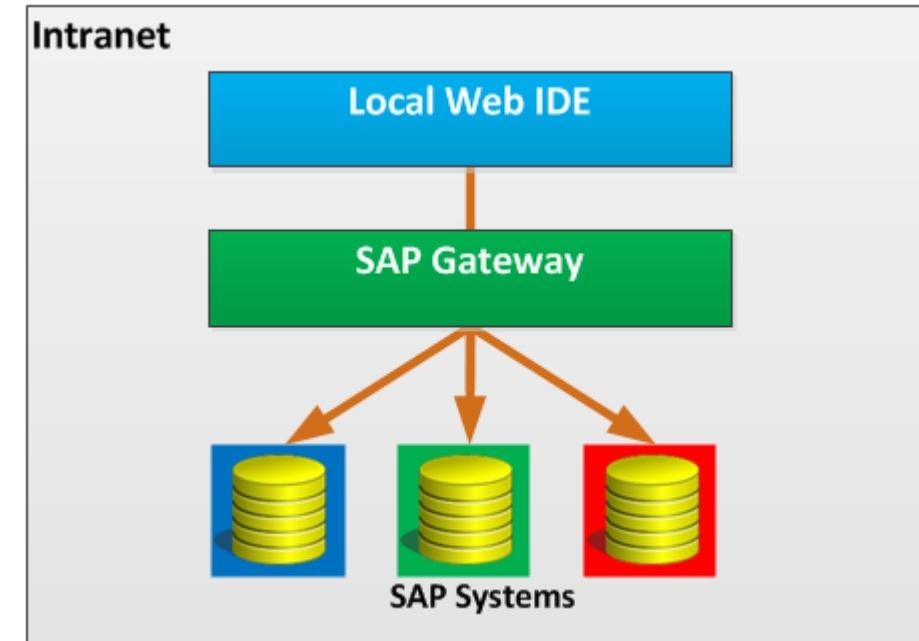
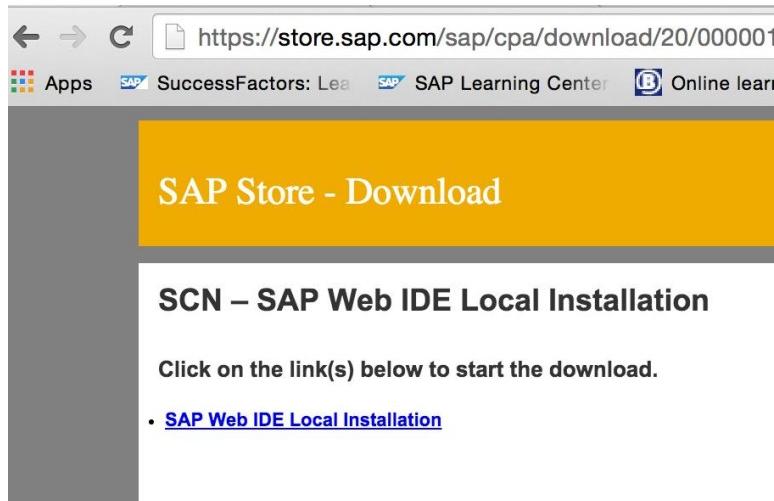
IDE Setup (cont.)

- Option 1: Cloud/SAP HCP
 - Pros
 - ▶ No maintenance required, always latest version
 - ▶ No license cost (for SAP system connections)
 - ▶ Fiori apps can be hosted in the cloud
 - ▶ Generates Component-preload.js
 - Cons
 - ▶ Requires SAP HANA Cloud Connector
 - ▶ May be slower compared to local installation
- Register at: <https://account.hanatrial.ondemand.com/>



IDE Setup (cont.)

- Option 2: Local Web IDE installation
 - Pros
 - ▶ Installed locally on Windows Local
 - ▶ Free, no licensing cost
 - Cons:
 - ▶ No updates so far after SAP Web IDE 1.12



Warning

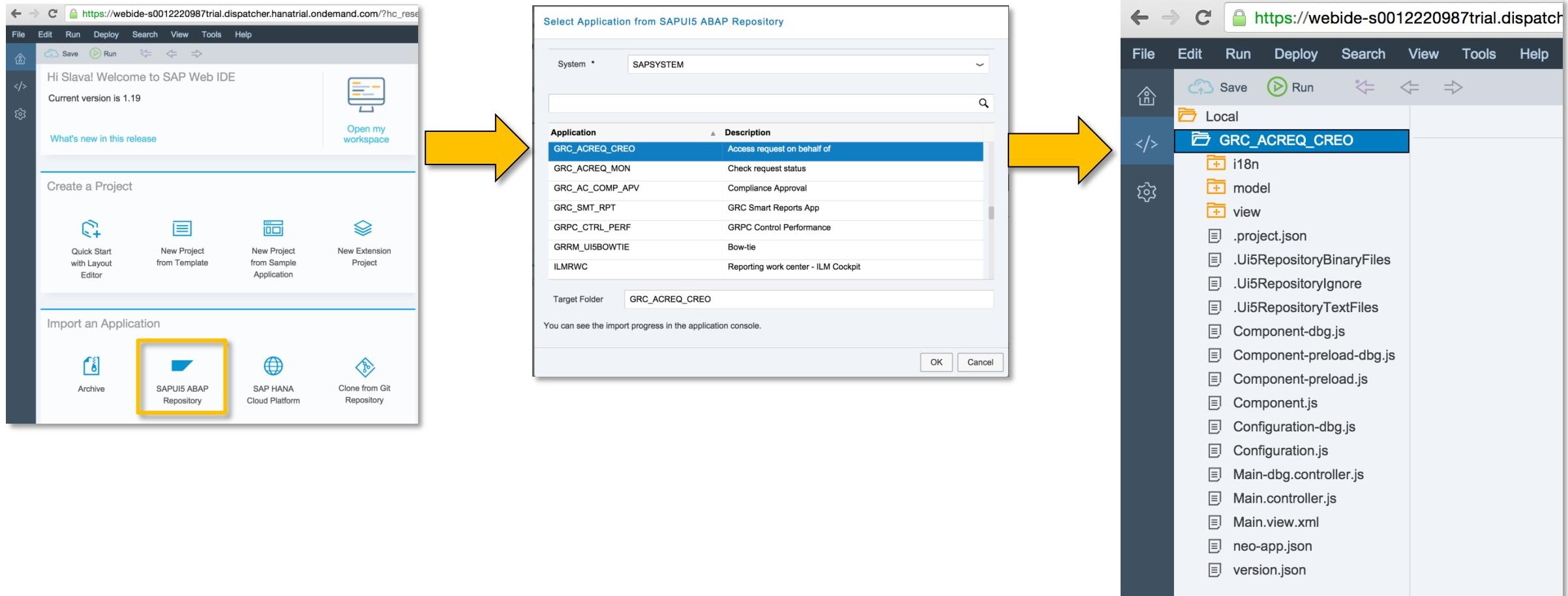
Do not use Fiori Toolkit for Eclipse, it is deprecated!

What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Extending Fiori App

- Import original Fiori app



Extending Fiori App (cont.)

- Create Extension project

The screenshot illustrates the process of creating an extension project for a Fiori application. It consists of three main parts:

- SAP Fiori Launchpad:** On the left, a screenshot of the SAP Fiori Launchpad shows the 'File' menu open. The 'Extension Project' option is highlighted with a yellow arrow.
- NEW EXTENSION PROJECT Dialog:** In the center, a screenshot of the 'NEW EXTENSION PROJECT' dialog titled '1 Original Application and Name'. It shows the 'Original Application' field set to '/GRC_ACREQ_CREO' and the 'Extension Project' field set to 'GRC_ACREQ_CREOExtension'. A second yellow arrow points from the 'Extension Project' menu in the Launchpad to this dialog.
- Fiori App Outline:** On the right, a screenshot of a Fiori application page ('Roles by Business Process') with an 'Outline' panel open. The 'Outline' panel is highlighted with a yellow border and shows the hierarchical structure of the app's views and controllers. The 'S2' view is selected. A third yellow arrow points from the 'Extension Project' dialog to this Fiori app.

Extending Fiori App (cont.)

- Extend only views and controllers that need to be changed
- For Business Case we extended:
 - View S2: adding search filters
 - Controller S2: fill in search filters
 - Controller RoleSearch: send filter criteria to GRC
 - Controller S4: hide extra custom fields

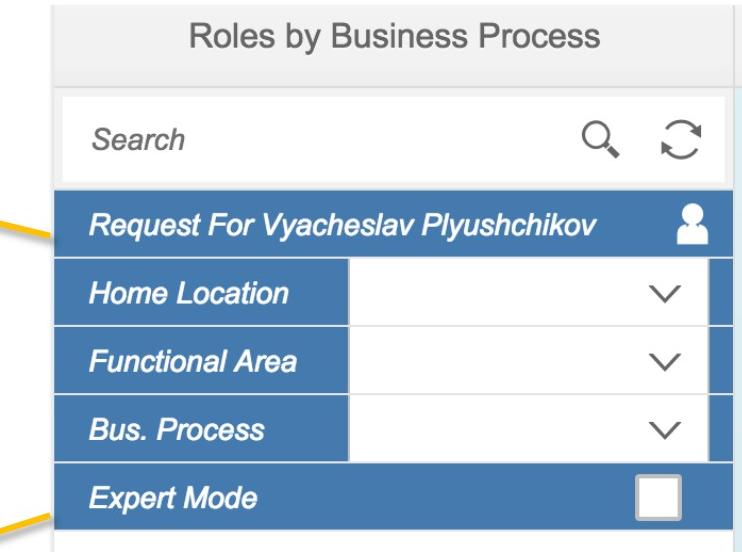
Outline

Show all elements ▾

- ▼ Views
 - ▶ Main
 - ▶ RoleSearch
 - ▶ **S2Custom (Replacement for S2)**
 - ▶ S3
 - ▶ S4
 - ▶ Welcome
- ▼ Fragments
- ▼ Controllers
 - Main
 - RoleSearchCustom (Extension for Roles)**
 - S2Custom (Extension for S2)**
 - S3
 - S4Custom (Extension for S4)**
 - Welcome

Extending Fiori App (cont.)

- Add Needed UI Elements to S2:



The screenshot shows the SAP Fiori Launchpad with the 'Roles by Business Process' application open. The application interface includes a search bar, a user profile icon, and four expandable sections: 'Home Location', 'Functional Area', 'Bus. Process', and 'Expert Mode'. A yellow arrow originates from the 'Expert Mode' section and points down to a code editor window.

```

Save Run ← →
Local
GRC_ACREQ_CREO
GRC_ACREQ_CREOExtension
  webapp
    localService
  view
    RoleSearchCustom.controller.js
    S2Custom.controller.js
  S2Custom.view.xml
S4Custom.controller.js
Component.js
index.html
.project.json
neo-app.json

*S2Custom.view.xml x
22      <core:Icon activeBackgroundColor="aqua" backgroundColor="" color="white" hoverBackgroundColor="#0000FF" press="initiate" src="sap-icon://person-placeholder"></core:Icon>
23      </content>
24    </Toolbar>
25
26
27    <Toolbar active="false" design="Info">
28      <content>
29        <Label id="SelectLoc" text="Home Location" textAlign="Right" width="90%"></Label>
30        <Select id="sLoc" width="170px"></Select>
31      </content>
32    </Toolbar>
33    <Toolbar active="false" design="Info">
34      <content>
35        <Label id="SelectJob" text="Functional Area" textAlign="Right" width="90%"></Label>
36        <Select id="sJob" width="170px"></Select>
37      </content>
38    </Toolbar>
39    <Toolbar active="false" design="Info">
40      <content>
41        <Label id="SelectBP" text="Bus. Process" textAlign="Right" width="90%"></Label>
42        <Select id="sBP" width="170px"></Select>
43      </content>
44    </Toolbar>
45    <Toolbar active="true" design="Info">
46      <content id="cont1">
47        <Label id="lException" text="Expert Mode" textAlign="Right" width="90%"></Label>
48        <CheckBox id="cException" select="onTOMEExceptionChange" selected="false"></CheckBox>
49      </content>
50    </Toolbar>
51
52      <List id="bprocList" items="/BusinessProcesses" mode="{device}/listMode" select="bprocHandleSelect" showNoData="false">

```

Extending Fiori App (cont.)

- Add logic to controller S2:

The screenshot shows the SAP Fiori Launchpad interface. On the left, the project structure is displayed under 'Local' with files like GRC_ACREQ_CREO, GRC_ACREQ_CREOExtension, webapp, localService, view, RoleSearchCustom.controller.js, S2Custom.controller.js (which is selected), S2Custom.view.xml, S4Custom.controller.js, Component.js, index.html, .project.json, and neo-app.json. In the center, the code editor shows the content of S2Custom.controller.js. The code defines three OData models: mModelLOC, mModelJOB, and mModelBP, each with specific configurations for data binding and aggregation.

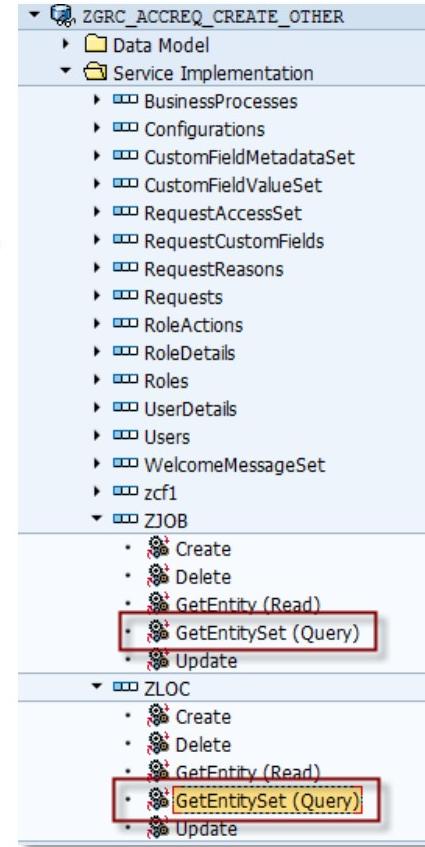
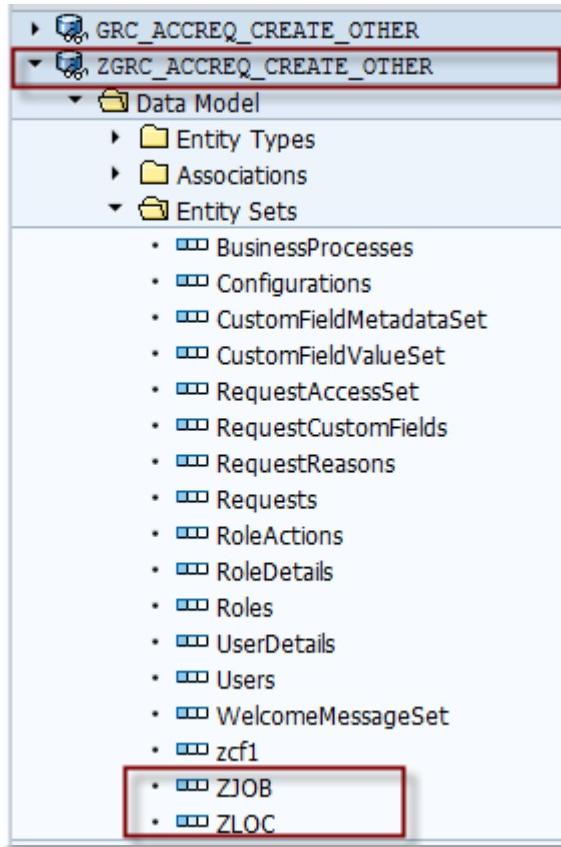
```
57 var mModelLOC = new sap.ui.model.odata.ODataModel("/sap/opu/odata/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV");
58 sap.ui.getCore().setModel(mModelLOC, "data_model");
59 var loctemplate = new sap.ui.core.Item({
60   key: "{Value}",
61   text: "{Text}"
62 });
63 var sloc = this.byId("sLoc");
64 sloc.setModel(sap.ui.getCore().getModel("data_model"));
65 sloc.bindAggregation("items", {
66   path: "/ZLOC",
67   length: 300,
68   template: loctemplate
69 });
70
71 var mModelJOB = new sap.ui.model.odata.ODataModel("/sap/opu/odata/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV");
72 sap.ui.getCore().setModel(mModelJOB, "jdata_model");
73 var sjob = this.byId("sJob");
74 sjob.setModel(sap.ui.getCore().getModel("jdata_model"));
75 sjob.bindAggregation("items", {
76   path: "/ZJOB",
77   length: 20,
78   template: loctemplate
79 });
80
81 var mModelBP = new sap.ui.model.odata.ODataModel("/sap/opu/odata/sap/GRC_ACCREQ_CREATE_OTHER");
82 sap.ui.getCore().setModel(mModelBP, "jdata_model");
83 var bptemplate = new sap.ui.core.Item({
84   key: "{BusinessProcessID}",
85   text: "{Description}"
86 });
87
```

Build custom “Z” OData Services where required

... or reuse standard OData Services

Extending OData Services

- Transaction SEGW
 - Copy original OData Service and extend the copy



```

Method ZLOC_GET_ENTITYSET
  call function 'DD_DOMVALUES_GET'
  exporting
    domname      = 'Z_USR_LOCGRP'
    text         = 'X'
    langu        = sy-langu
  tables
    dd07v_tab   = lt_dd07
  exceptions
    wrong_textflag = 1
    others       = 2.

  loop at lt_dd07 into ls_dd07.

    ls_entityset-custom_fieldname = ls_dd07-domvalue_1
    ls_entityset-value = ls_dd07-domvalue_1.
    ls_entityset-text = ls_dd07-ddtext.

    append ls_entityset to lt_entityset.
  endloop.

  sort lt_entityset.
  delete adjacent duplicates from lt_entityset.
  et_entityset[] = lt_entityset[].

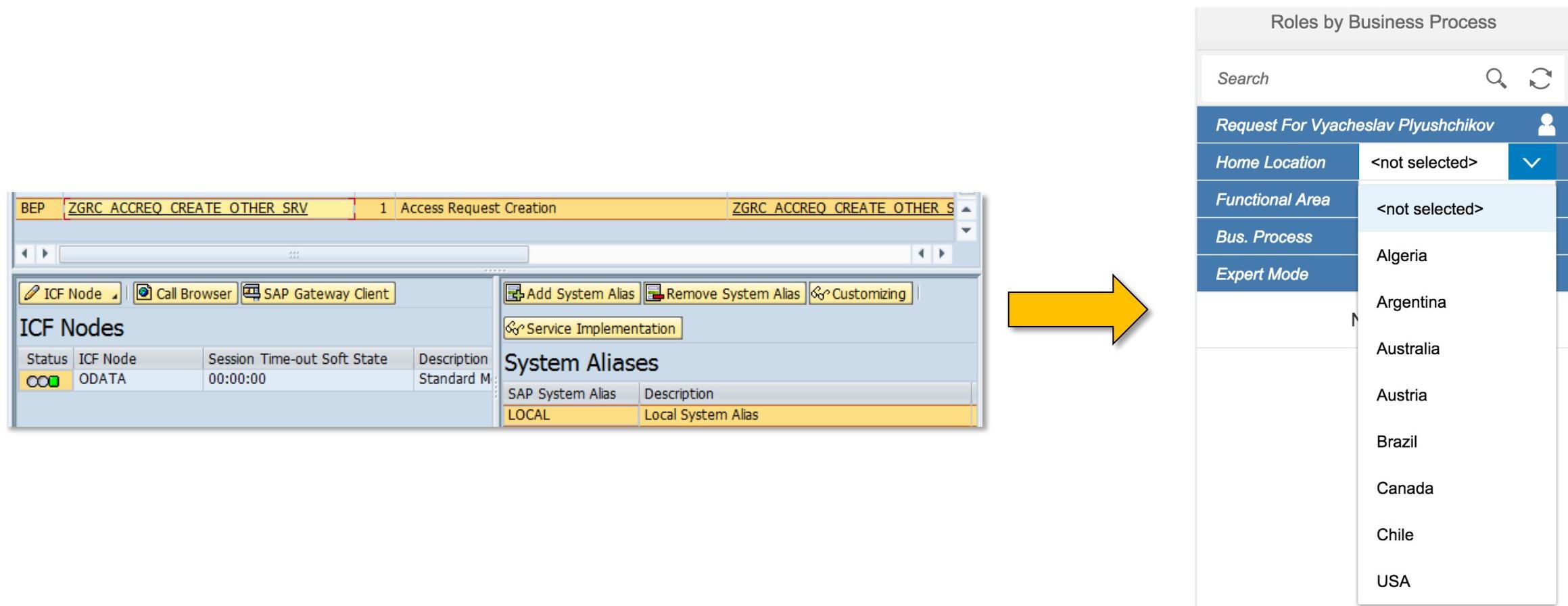
endmethod.

```

The screenshot shows the ABAP code for the ZLOC_GET_ENTITYSET method. The code implements a loop to process data from lt_dd07 and append it to lt_entityset. Two specific lines of code are highlighted with a red border: 'append ls_entityset to lt_entityset.' and 'et_entityset[] = lt_entityset[].'

Extending OData Services (cont.)

- Transaction /WFND/MAINT_SERVICE
 - Register OData service, maintain ICF node and system alias

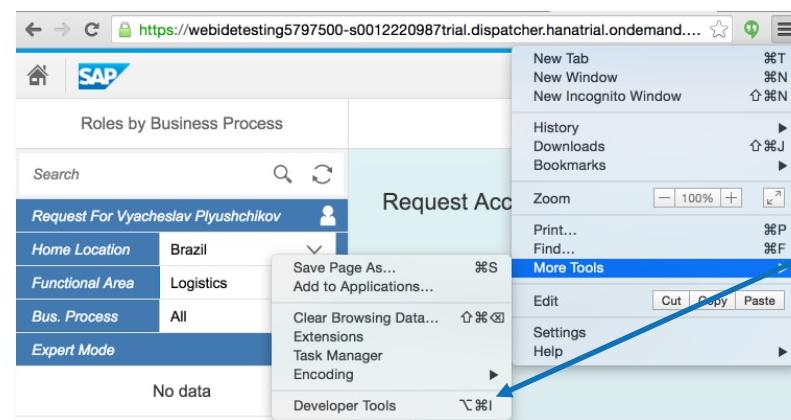


What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Debugging in Web Browser

- Chrome is recommended for HTML5/Fiori development



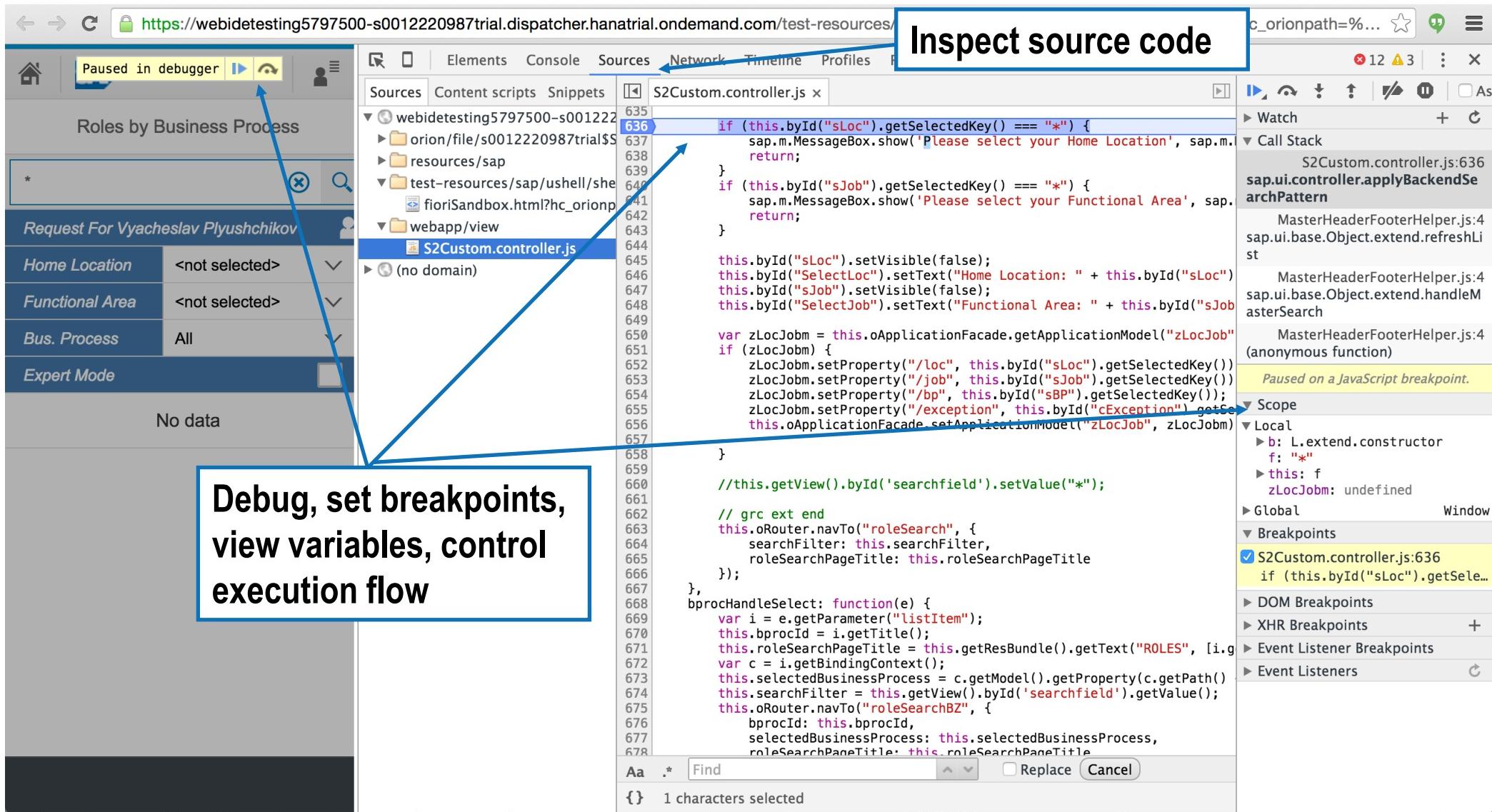
Start developer tools in the browser

The screenshot shows the SAP Fiori sandbox application within a browser. The developer tools are open, with the 'Elements' tab selected. A blue box highlights the 'Inspect UI elements' callout. A blue arrow points from the 'Elements' tab to this callout. Another blue arrow points from the highlighted element in the UI to the corresponding DOM node in the developer tools panel. A third blue arrow points from the developer tools panel to a callout box containing two bullet points: 'Automatic highlight' and 'Access to properties'.

Inspect UI elements

- Automatic highlight
- Access to properties

Debugging in Web Browser (cont.)



Tracing OData Calls

- User SAP Gateway tools
 - /IWFND/ERROR_LOG
 - ▶ Gateway error log: records errors for calls to SAP GW
 - /IWFND/TRACES
 - ▶ Gateway call traces: performance and payload traces
 - /IWFND/GW_CLIENT
 - ▶ SAP Gateway Client: analyze and repeat calls

Gateway Error Log

- First log to check if calls are failing

The screenshot shows the SAP Gateway Error Log interface. The main window displays an 'Overview' table of errors. The first error row is selected, highlighting the date (12.01.2016) and time (17:49:44). The error text indicates a service not found for namespace GRC_ACREQ_CREATE_OTHER_SRV. A blue arrow points from the 'Review error log' callout to this error text. Another blue arrow points from the 'Analyze the call, replay if required' callout to the 'Call Stack' button in the toolbar below the overview table.

SAP Gateway: Error Log

Overview

Line	Entr...	Date	Time	T100 Error ID	T10...	Error ...	ICF Node	B	Error Text
3	1	12.01.2016	17:49:44	/IWFND/MED170		1		<input type="checkbox"/>	No service found for namespace , name GRC_ACREQ_CREATE_OTHER_SRV, version 0001
2	1		17:49:27	/IWBEPC/CM_MGW_RT021		odata		<input checked="" type="checkbox"/>	Method 'BUSINESSPROCESSE_GET_ENTITYSET' not implemented in data provider class
1	1		17:49:01	/IWFND/CM_CONSUMER122		odata		<input type="checkbox"/>	Invalid system query options value

Error Context

Ex...	Name	Value
	.ERROR_CONTEXT	
	.ERROR_INFO	No service found for namespace , name GRC_ACREQ_CREATE_OTHER_SRV, version 0001
	.ERROR_RESOLUTION	
	...SAP_NOTE	See SAP Note 1797736 for error analysis
	...LINK_TO_SAP_NOTE	https://service.sap.com/sap/support/notes/1797736
	/IWFND/MED_MPL_ACCESS	

Review error log

Analyze the call,
replay if required

Gateway Traces

- Analyze OData calls
- Review performance trace
- Replay calls to troubleshoot/debug issues

SAP Gateway: Tracing Tools

Error Log

Configuration Performance Trace Payload Trace

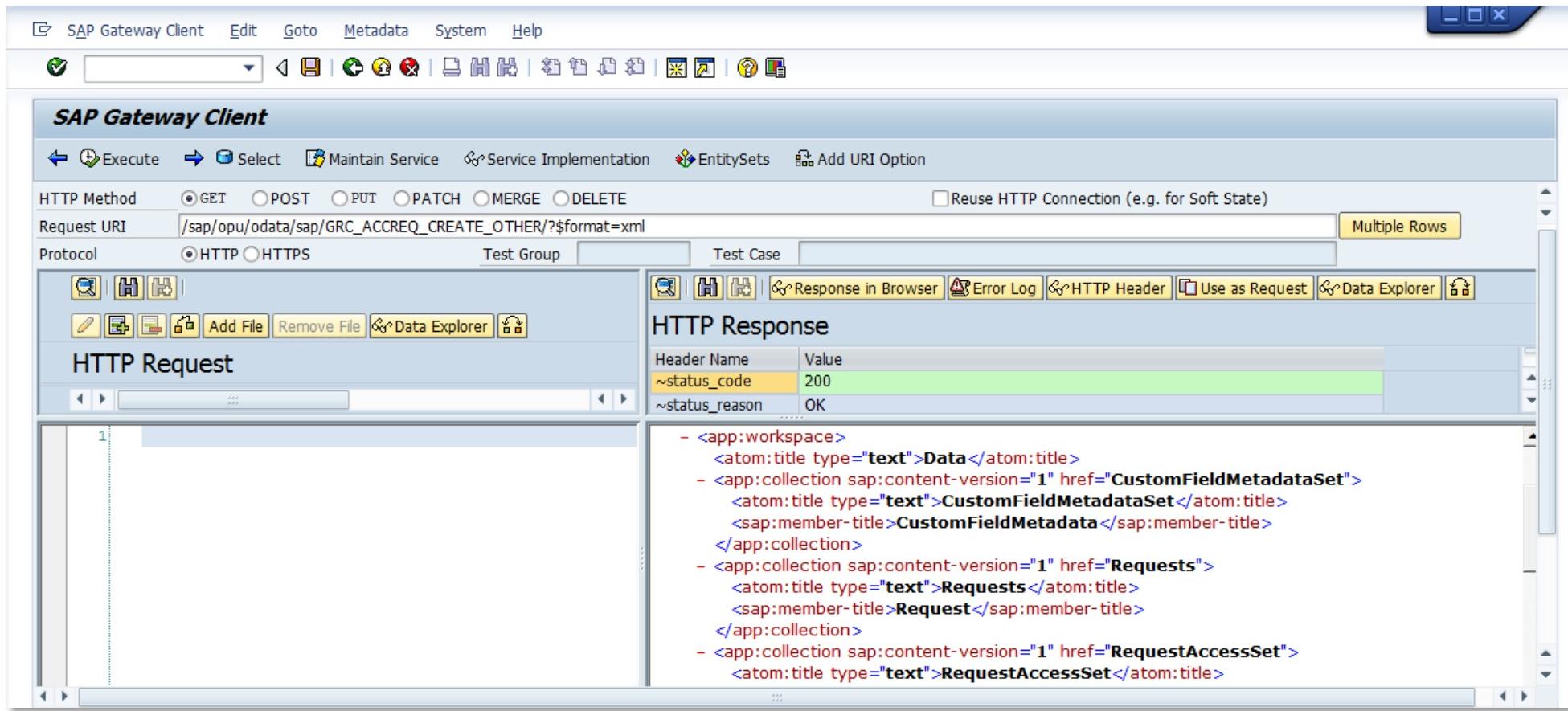
Users & Request URI Prefix
TST-GRC-USR2

Payload Trace: Client 100 User TST-GRC-USR2

Service Call Info	Method	Proc. Time	Appl. Time	Req. Size	Resp. Size	Format	Date	Time	Expiry Date	Status
/sap/GRC_ACCREQ_CREATE_OTHER/BusinessProcesses?\$skip...	GET	350	281	0	4.490	json	16.01.2016	12:04:18	30.01.2016	green
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	131	31	0	1	txt	16.01.2016	12:04:18	30.01.2016	green
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	132	33	0	1	txt	16.01.2016	12:04:18	30.01.2016	green
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	146	0	0	953	xml	16.01.2016	12:04:17	30.01.2016	red
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	149	34	0	1	txt	16.01.2016	12:04:17	30.01.2016	green
/sap/GRC_ACCREQ_CREATE_OTHER/BusinessProcesses?\$skip...	GET	364	283	0	4.490	json	16.01.2016	12:04:17	30.01.2016	green
/sap/ZGRC_ACCREQ_CREATE_OTHER_SRV/BusinessProcesses...	GET	172	0	0	953	xml	16.01.2016	12:04:17	30.01.2016	red
/sap/GRC_ACCREQ_CREATE_OTHER/Users?\$filter=UserID%20...	GET	182	94	0	900	json	16.01.2016	12:04:06	30.01.2016	green
/sap/GRC ACCREQ CREATE OTHER/WelcomeMessageSet?\$e	GET	134	57	0	5.159	json	16.01.2016	12:04:01	30.01.2016	green

Gateway Client

- Retrieve OData services metadata
- Execute OData calls, review responses, troubleshoot issues

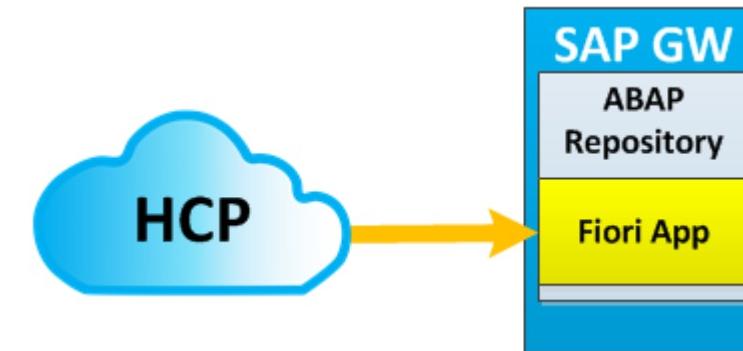


What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Deploying Fiori App

- Deploy app code to ABAP system
 - Direct Deployment to SAPUI5 ABAP Repository
 - Export/Import to SAPUI5 ABAP Repository
- Give users access to Fiori app
 - Navigation Design
 - User Permissions



Direct Deployment to ABAP Repository

- Direct Deployment to SAPUI5 ABAP Repository

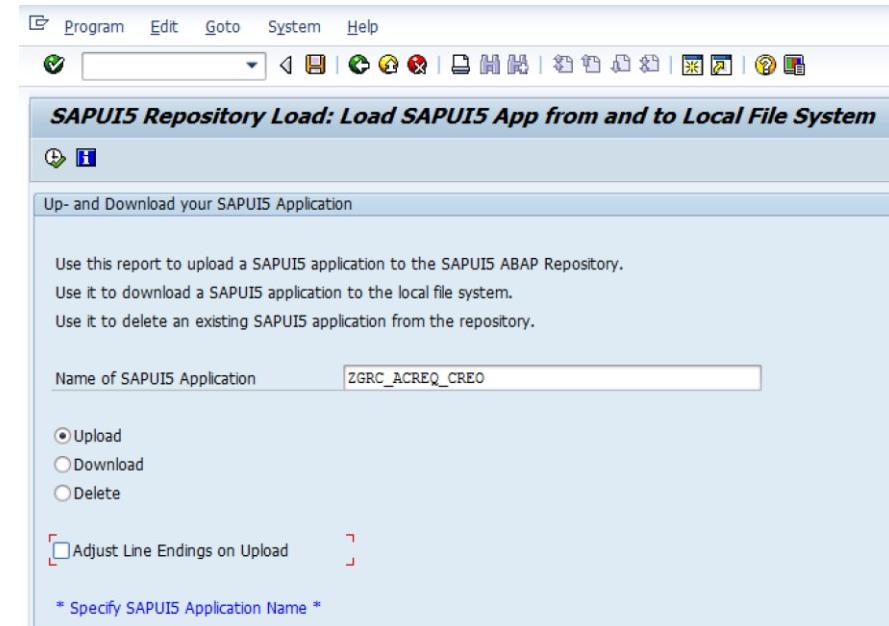
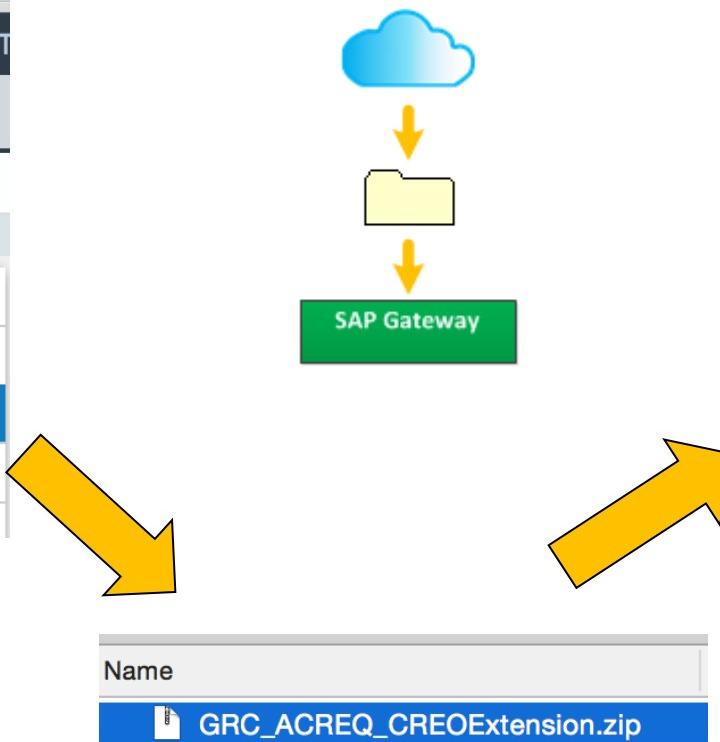
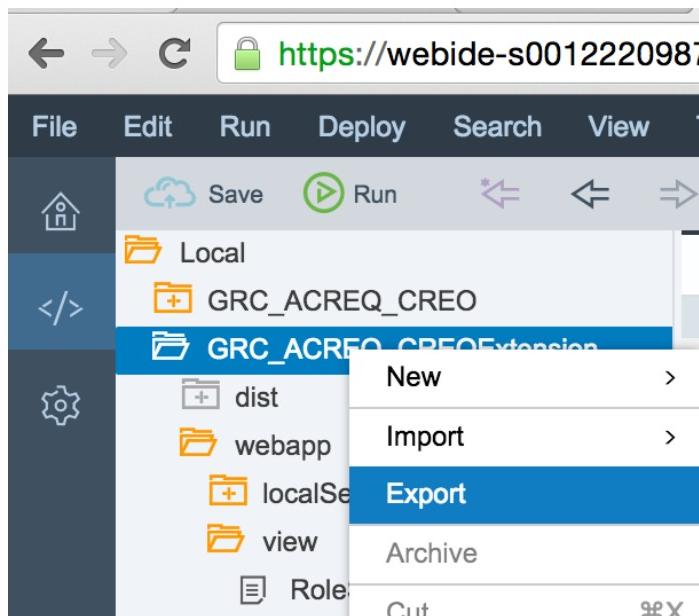
The screenshot illustrates the deployment process for a new SAPUI5 application to an ABAP repository. It consists of three sequential steps:

- Step 1: Deployment Options**
 - System: SAPSYSTEM
 - Deploy a new application
 - Update an existing application
- Step 2: Deploy a New Application**
 - Name: ZGRC_ACREQ_CREO
 - Description: ZGRC_ACREQ_CREO
 - Package: ZGRAC
- Step 3: Create a new request**
 - Create a new request
 - Request Description: New Fiori app
 - The Request ID is automatically generated

Yellow arrows indicate the progression from Step 1 to Step 2, and from Step 2 to Step 3.

Export/Import to ABAP Repository

- Export from HCP
- Import via /UI5/UI5_REPOSITORY_LOAD report



Provide Access to Fiori App

- Define Semantic Object /UI2/SEMOBJ
- Define app alias in LPD_CUST



New Entries: Overview of Added Entries

Semantic Object	Semantic Object Name	Semantic Object Description
ZAccessRequest	ZAccessRequest	Extended Access Request for Other

Change Launchpad - Role: UIGRC001 Instance: TRANSACTIONAL (EN)

Customized Version

- GRC: Transactional Apps
 - Inactive Applications
 - Check Request Status
 - Request access
 - Access Request
 - Analyze Risks
 - Maintain Role
 - Analyze Users
 - Analyze Roles
 - Analyze Risks
 - Approve Access Request
 - Smart Reports
 - Request access for others
 - Z Request access for other**
 - Compliance approver

Link Details

Link Text: **Z Request access for other**

Application Type: URL

URL: `/sap/bc/ui5_ui5/sap/zgrc_acreq_crea...`

System

System Alias:

Advanced Parameters (Optional)

Application - Deactivation by User: Application Cannot be Removed from Launchpad

Application-Related Parameters

Application Alias: **ZACCESSREQUESTO**

Target App. Parameters:

Suspend/Resume:

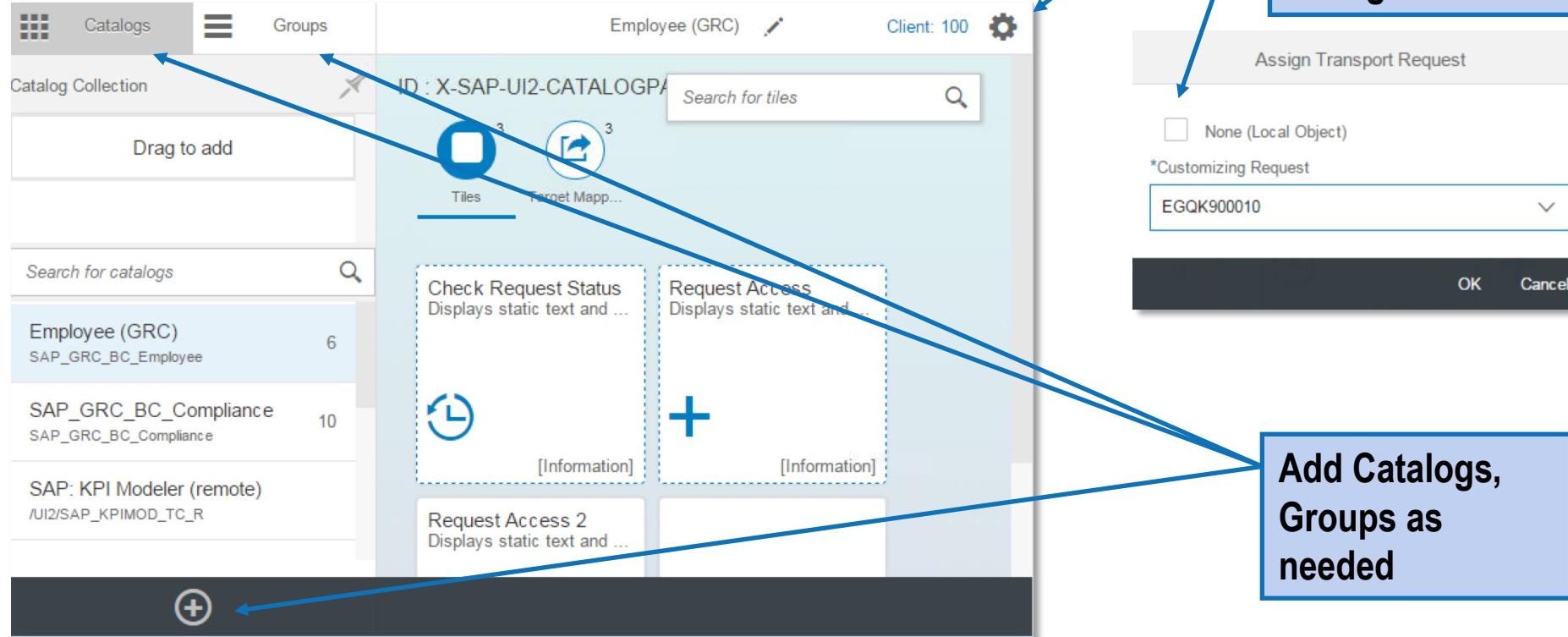
Proxy Class:

Additional Information: `SAPUI5.Component=fcg.grc.acreq.creat...`

FPM Event ID:

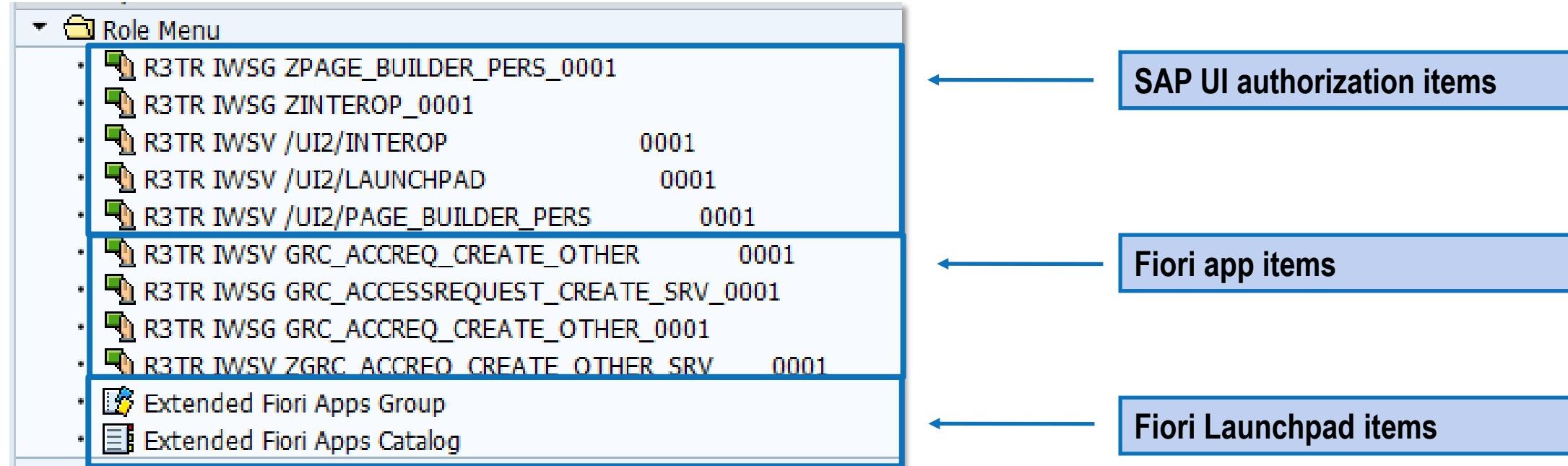
Provide Access to Fiori App (cont.)

- Define Catalogs and Groups
 - /sap/bc/ui5_ui5/sap/arsvc_upb_admn/main.htm
- Create tiles, target mappings, add tiles to groups



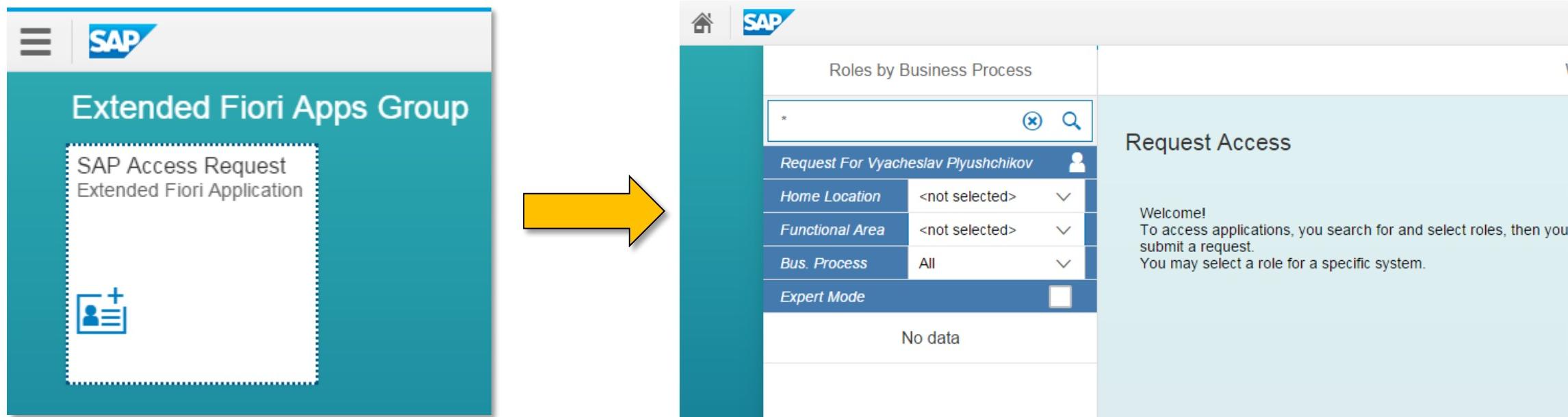
Provide Access to Fiori App (cont.)

- Define PFCG roles for Fiori Launchpad
 - Define access to catalogs, groups, OData services, required Fiori Launchpad services



Provide Access to Fiori App (cont.)

- Assign new role to users
- User can see new tile in Fiori Launchpad



Demo: Modifying and Deploying Fiori App

SAP Web IDE in Action



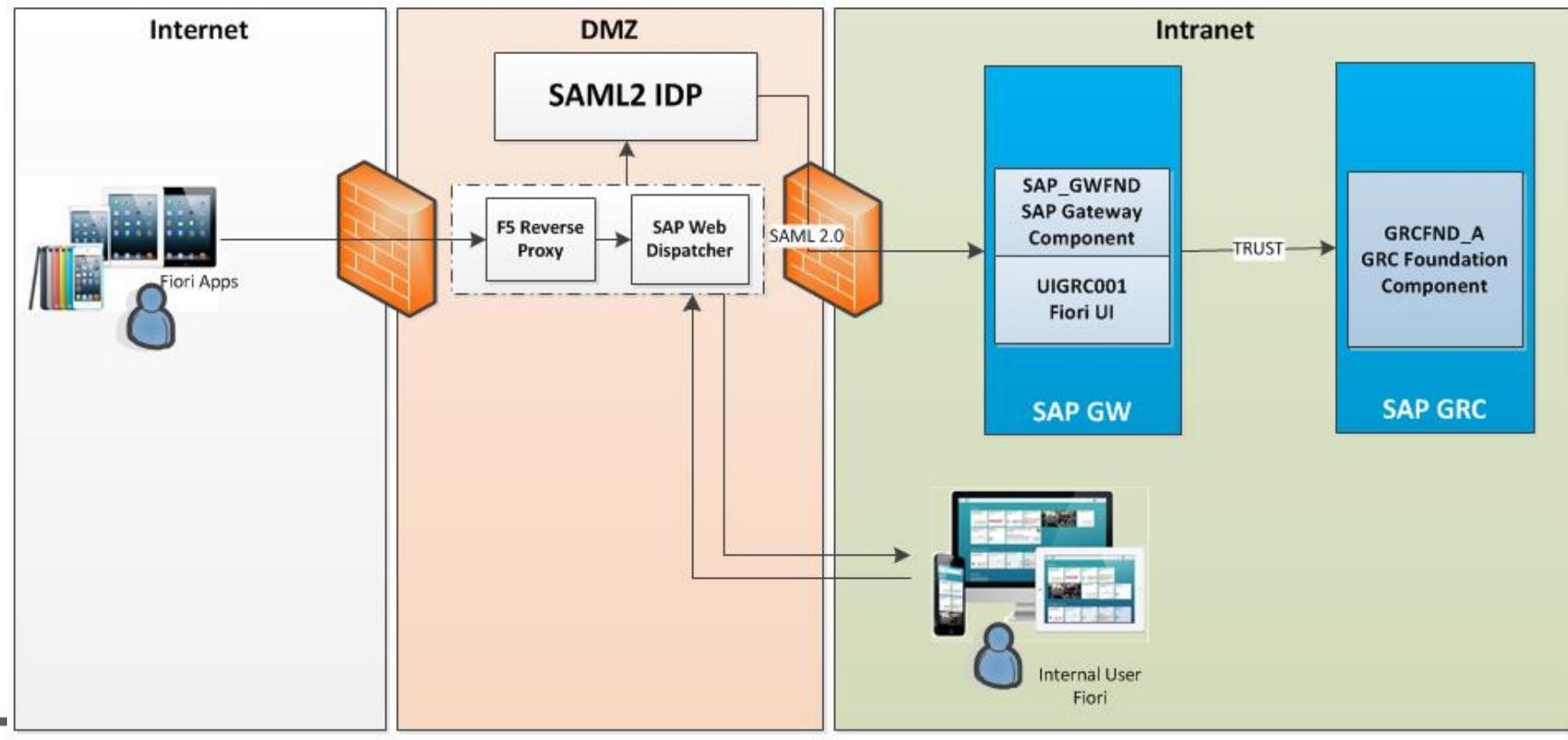
Demo

What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Single Sign-On

- Most common solution is based on SAML2
 - Uses HTTPS ports
 - Supports identity federation



Branding

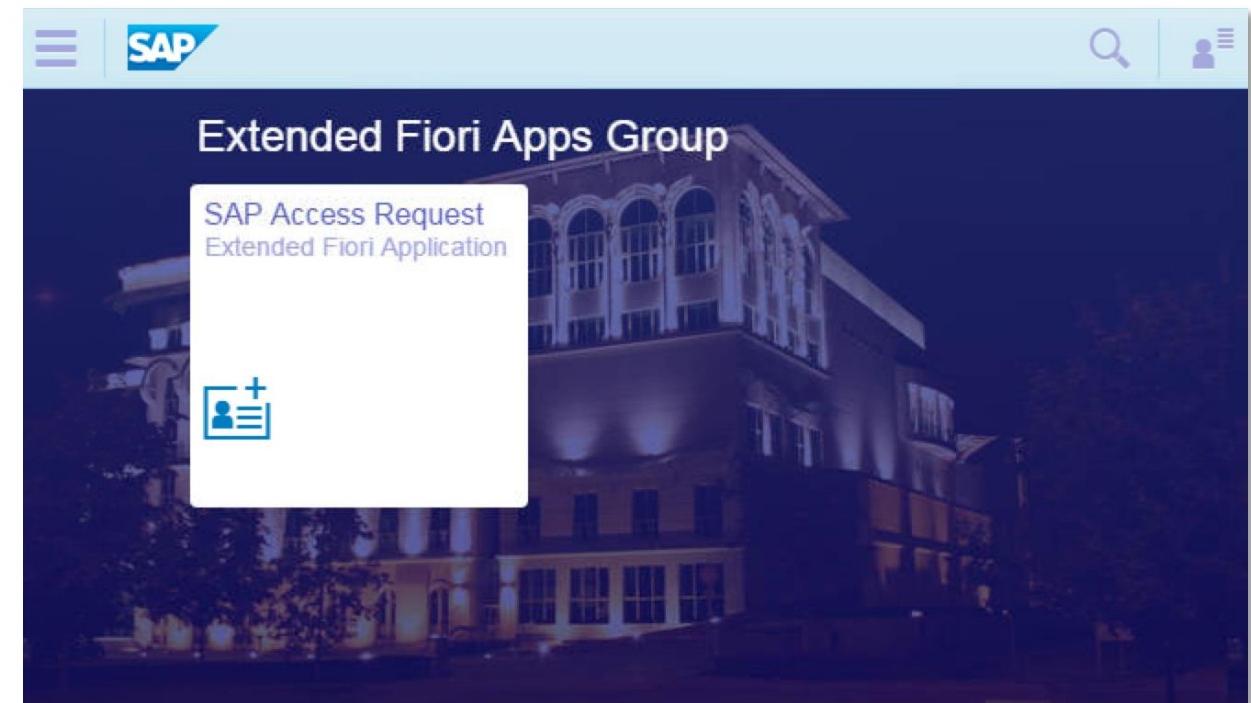
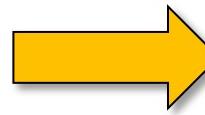
- Theme Designer
 - /ui5/theme_designer
 - Copy existing theme to a new name and customize

Expert

Filters: View Options:

Theme Parameter	Value
sapBackgroundColor	#f2f2f2
sapBackgroundGradientBaseColor	rgba(18,19,114,0.71)
sapBackgroundImage	<uploaded image>
sapBackgroundImageOpacity	1.0
sapBackgroundImageRepeat	true
sapBaseColor	#2D8EB5
sapBrandColor	#009de0
sapButton_Accept_Background	@sapPositiveElementColor
sapButton_Background	#f7f7f7
sapButton_BorderColor	darker(@sapButton_Background)
sapButton_BorderRadius	0

CSS



What We'll Cover

- GRC Access Control Fiori overview
- Business scenario for Fiori app extension
- Implementation: IDE setup
- Implementation: Fiori extension project
- Tracing and debugging
- Deployment
- Tips and tricks
- Wrap-up

Where to Find More Information

- <http://help.sap.com/grc-ac#section6>
 - GRC Access Control: SAP Fiori apps documentation on the SAP Help Portal
- <https://account.hanatrial.ondemand.com>
 - SAP HANA Cloud Platform: registration, access to management cockpit
- <https://sapui5.hana.ondemand.com>
 - SAP UI Development Toolkit: SAPUI5/Fiori development
- <http://help.sap.com/nwgateway#section7>
 - SAP Gateway 2.0: SAP GW/OData services development on the SAP Help Portal
- <http://advanced-view.com>
 - Vyacheslav Plyushchikov's blogs on GRC Access Control and Fiori apps

7 Key Points to Take Home

- SAP Fiori apps for GRC AC can easily be extended
- SAP provides free HCP-based IDE for developing and extending Fiori apps
- Use design thinking approach: start from “better future” goal, not from analyzing current problems
- Reuse as much as possible, change only app parts, where new functionality is needed
- Keep the app “lightweight” to load faster on mobile devices, no big images or large data pre-loads
- Build Fiori apps for access from the Internet: secure it with HTTPS, web filters, identity federation authentication
- Use trace and debug tools for front-end and back-end issue resolution

Your Turn!



Questions?

How to contact me:

**Vyacheslav (Slava) Plyushchikov
slava.plyushchikov@gmail.com**

Please remember to complete your session evaluation

Disclaimer

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. Wellesley Information Services is neither owned nor controlled by SAP SE.



Wellesley Information Services, 20 Carematrix Drive, Dedham, MA 02026
Copyright © 2016 Wellesley Information Services. All rights reserved.